



Team Fugu and the OpenBSD Tools Project



for the United States
Geological Survey
Astrogeology Division



Team Fugu Members

<http://www.cet.nau.edu/~fugu/>

- **Ben Atkin**
 - Communicator and Researcher
- **Thad Boyd**
 - Facilitator and Webmaster
- **Nauman Qureshi**
 - Recorder and Documenter
- **Erik Wilson**
 - Team Leader and Organizer

The OS Tools Project

Our client is migrating many of their systems to OpenBSD, because they have found it to be a better match for their needs in many areas of work.

Until now, however, they were missing two tools available on other systems:

- an **Automated Installer** and
- an **Automated Patcher**

These tools have been holding them back somewhat in their adoption of OpenBSD. This is where we come in. This semester, we have built these tools by extending already existing projects to meet our client's requirements. Using these tools, the administrators will save time, keep their systems more consistent, and keep security patches up-to-date.



<http://www.usgs.gov/>

Our Client

Ernest Bowman-Cisneros and Margaret Johnson
(System Administrators)

USGS Astrogeological Division - Flagstaff, AZ

Our client works at one of the largest research stations in the country for space exploration and mapping. They have participated in many high-profile projects including the Spirit and Opportunity rovers. Their space photography, mapping, and research efforts make heavy demands on computing infrastructure. To manage a large number of servers, they have learned to work efficiently and make use of the latest and greatest system administration tools.

About OpenBSD OpenBSD is a free UNIX variant, comparable to Linux, but with designed with security as the main goal. In fact, other free UNIX distributions use security tools that are part of the OpenBSD project, most notably OpenSSH.

- *OpenBSD* runs on many different hardware platforms.
- *OpenBSD* is thought of by many security professionals to be the most secure UNIX-like operating system as the result of a never-ending comprehensive source code security audit.
- *OpenBSD* is a full-featured UNIX-like operating system available in source form at no charge.
- *OpenBSD* integrates cutting-edge security technology suitable for building firewalls and private network services in a distributed environment.
- *OpenBSD* benefits from strong ongoing development in many areas, offering opportunities to work with emerging technologies with an international community of programmers and users.



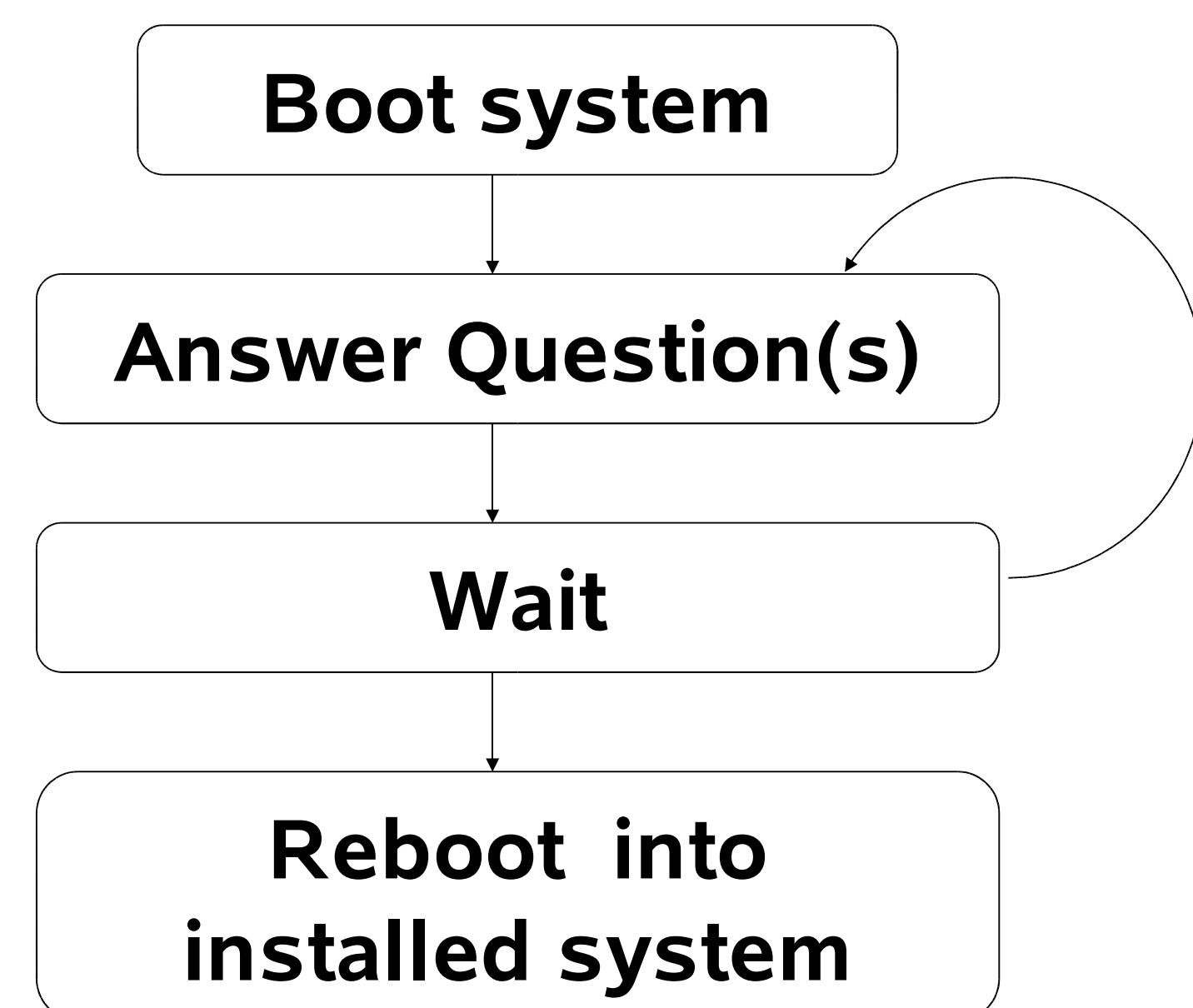
<http://www.openbsd.org/>

Automated Installer

Overview

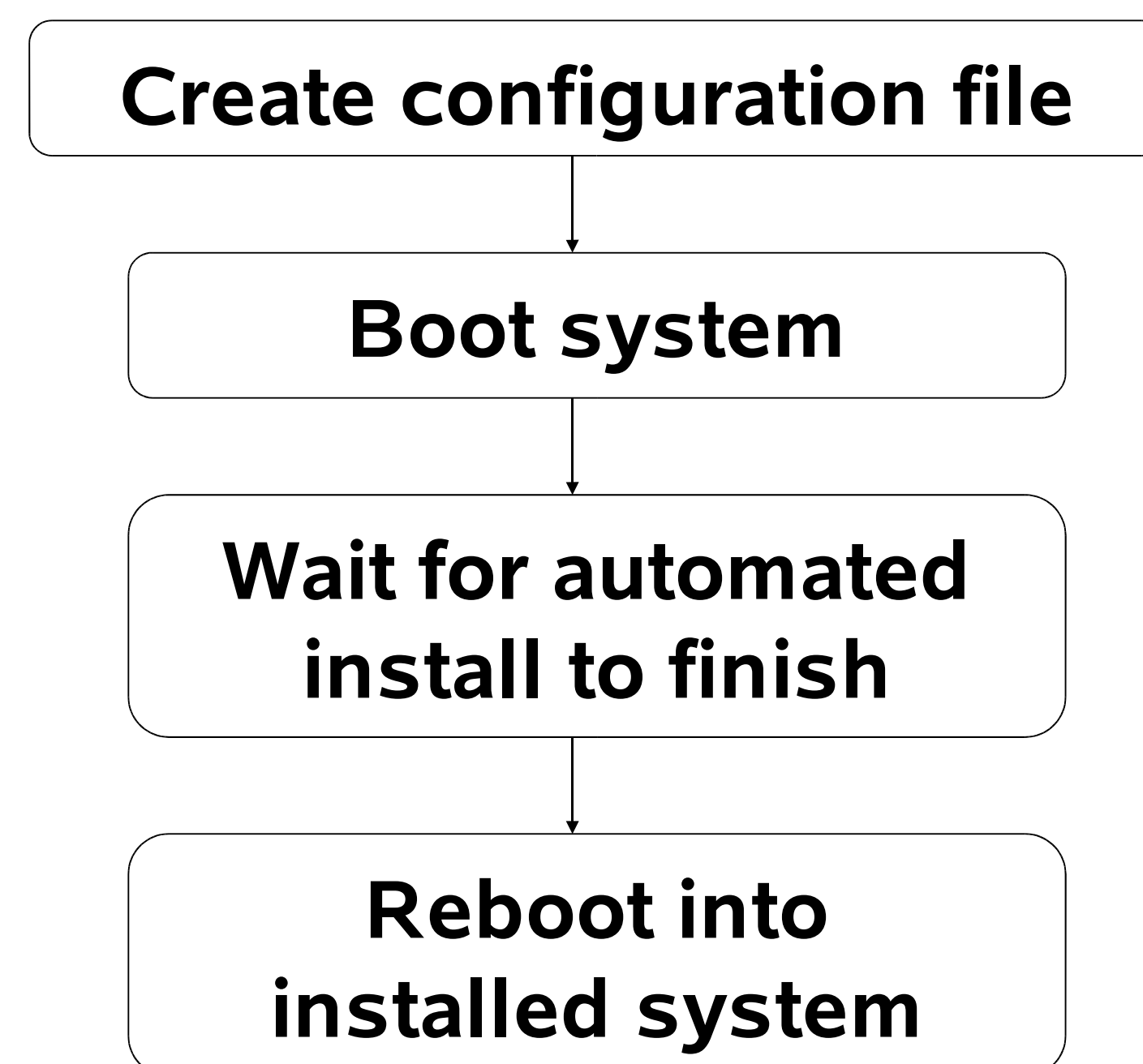
- Based on existing OpenBSD installer
- Operates automatically, greatly increasing efficiency

Standard (Interactive) Installer



=20+ minutes/computer

Automated Installer



=10 minutes work for a group of computers

- Reads configuration options from a file

Configuration File Sections

General	Behavior, Pre- and Post-Install Scripts
Network	Network Configuration (hostname, address, subnet)
Disks	Disk Geometry
Filesets	Program archives to download and install
Final	Final configuration settings such as time zone

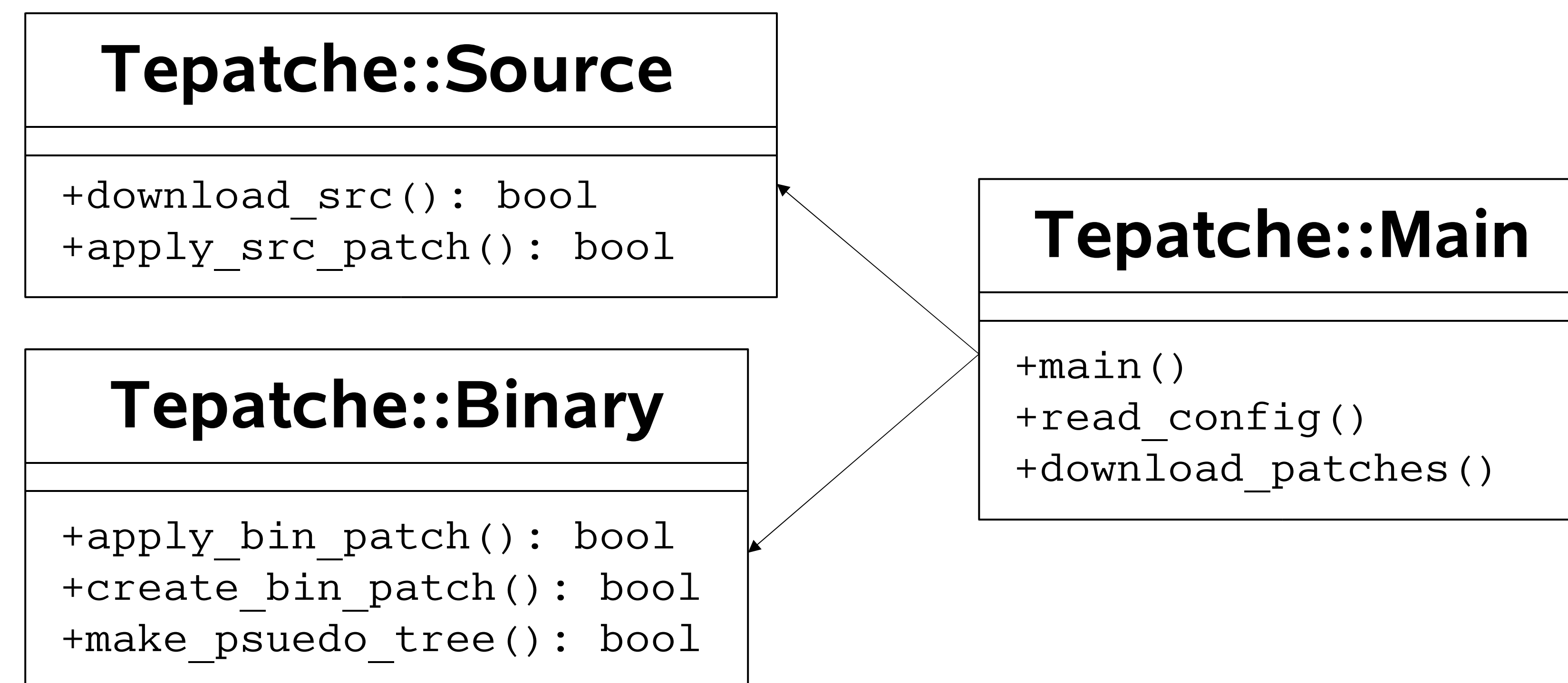
Features

- Single configuration file can be applied to systems with different brands of hardware and different disk geometry
- File can be loaded from disk or network
- User can interrupt installer and override option specified in the configuration file
- Internet monitoring available

Automatic Patcher

Overview

- Automatic patcher downloads and installs security updates
- Security updates are released as issues are discovered
- Important part of a secure system



Work Completed

- Added features to Tepatche project created by Gunnar Wolf at UNAM (Universidad Nacional Autonoma de México)
 - Ability to create binary patches
 - Ability to install binary patches
- Split patcher into modules (diagram above)

Binary Patching

Before we added binary patching, all that was available was source patching. Source patching requires that patches be compiled into machine code on each computer. Compiling can take the system hours of time. While it's compiling, the full performance of the computer is not available. Binary patching allows the code to be compiled once, on an administrator computer, and then transferred in binary (machine code) form to the other computers. This helps keep the researchers' servers and workstations to be responsive at all times.